



# UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/816,887	03/23/2001	William A. Pugh	109870-130111	6896

25943 7590 07/12/2005

SCHWABE, WILLIAMSON & WYATT, P.C.  
PACWEST CENTER, SUITE 1900  
1211 SW FIFTH AVENUE  
PORTLAND, OR 97204

EXAMINER
----------

CHOJNACKI, MELLISSA M

ART UNIT	PAPER NUMBER
----------	--------------

2164

DATE MAILED: 07/12/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/816,887

Applicant(s)

PUGH, WILLIAM A.

Examiner

Melissa M. Chojnacki

Art Unit

2164

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 18 April 2005.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-38 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-38 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

## DETAILED ACTION

### Remarks

1. In response to communications filed on April 18, 2005, no claims are cancelled or amended, and no new claims have been added. Therefore, claims 1-38 are still presently pending in the application.

### ***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1, 3-5, 9, 11, 13-15 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Byrne et al. (U.S. Patent No. 6,823,338) in view of Shadmon (U.S. Patent No. 6,208,993).

As to claim 1, Byrne et al. teaches a method for copying/archiving a web based application (See abstract; column 1, lines 15-38), the method comprising:

initializing a file to store the web based application, including creation of a root directory within the file (See abstract; Fig. 2; column 1, lines 54-57; column 3, line 39);

and initializing a first plurality of storage data objects under the data directories for all non-file system structures of the web based application (See abstract; column 2, lines 48-62, where "non-file system structures" is read on "tables"; column 3, lines 44-46); and

Art Unit: 2164

copying and storing the non-file system structures into the first plurality of storage data objects (See abstract; Fig. 2 and Fig. 3; column 2, lines 48-62, where "non-file system structures" is read on "table"; column 3, lines 44-46; column 4, lines 53-62).

Byrne et al. does not teach creating data directories under the root directory.

Shadmon teaches a method and system for uniformly accessing multiple directory services (See abstract), in which he teaches creating data directories under the root directory (See Fig. 1; column 38; lines 9-20, where "data dictionary" is read on "data directories").

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al., to include creating data directories under the root directory.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al., by the teachings of Shadmon because creating data directories under the root directory would maintain a so-called balanced structure of the tree index in order to avoid long paths for accessing a given data record from the root node to the leaf node that is associated with the sought data record (See Shadmon column 4, lines 38-42).

As to claims 3 and 13, Byrne et al. as modified, teaches wherein the creating of data directories under the root directory (See Shadmon, Fig. 1; column 38; lines 9-20) and initializing a first plurality of storage data objects under the data directories

Art Unit: 2164

comprises creating an application level data directory under the root directory (See Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62; also see Shadmon, Fig. 1; column 38; lines 9-20);

wherein the programming instructions, when executed, operate the apparatus to create an application level data directory under the root directory to create data directories under the root directory (See Shadmon, Fig. 1; column 38; lines 9-20) and initialize a first plurality of storage data objects under the data directories (See Shadmon, Fig. 1; column 38; lines 9-20; also see Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62).

As to claims 4 and 14, Byrne et al. as modified, teaches wherein the creating of data directories under the root directory (See Shadmon, Fig. 1; column 38; lines 9-20) and initializing a first plurality of storage data objects under the data directories (See Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62; also see Shadmon, Fig. 1; column 38; lines 9-20) further comprises:

initializing a first of the first plurality of storage data objects under the application level data directory to store a structural description describing non-file system structures and files of a file system of the web based application (See Byrne et al., abstract; column 2, lines 48-62, where "non-file system structures" is read on "tables"; column 3, lines 44-46); and

Art Unit: 2164

copying and storing the structure description in the first of the first plurality of storage data objects (See Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62);

wherein the programming instructions, when executed, operate the apparatus to:

initialize a first of the first plurality of storage data objects under the application level data directory to store a structural description describing non-file system structures and files of a file system of the web based application (See Byrne et al., abstract; column 2, lines 48-62, where “non-file system structures” is read on “tables”; column 3, lines 44-46), and

copy and store the structure description in the first of the first plurality of storage data objects (See Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62).

As to claims 5 and 15, Byrne et al. as modified, teaches wherein the copying and storing of non-file system structures into the first plurality of storage data objects comprises:

initializing a second of the first plurality of storage data objects under the application level data directory to store a user description describing users of the web based application (See Byrne et al., abstract; column 2, lines 48-62, where “non-file system structures” is read on “tables”; column 3, lines 44-46); and

copying and storing the user description in the second of the first plurality of storage data objects (See Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62);

wherein the programming instructions, when executed, operate the apparatus to initialize a second of the first plurality of storage data objects under the application level data directory to store a user description describing users of the web based application (See Byrne et al., abstract; column 1, lines 22-26; column 2, lines 48-62, where "non-file system structures" is read on "tables"; column 3, lines 44-46; column 5, lines 41-53), and to copy and store the user description in the second of the first plurality of storage data objects (See Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62).

As to claims 9 and 19, Byrne et al. as modified, teaches wherein the method further comprises copying and storing files of the web based application that are part of a file system into the file for storing the web based application as second plurality of storage data objects under the root directory (See Byrne et al., abstract; column 2, lines 48-62, where "non-file system structures" is read on "tables"; column 3, lines 44-46); wherein the programming instructions, when executed, operate the apparatus to copy and store files of the web based application that are part of a file system into the file for storing the web based application as second plurality of storage data objects under the root directory (See Byrne et al., abstract; Fig. 2 and Fig. 3; column 2, lines 48-62;

Art Unit: 2164

column 3, lines 44-46; column 4, lines 53-62; also see Shadmon, Fig. 1; column 38; lines 9-20).

As to claim 11, Byrne et al. teaches an apparatus comprising:

storage medium having stored therein programming instructions (See abstract; column 14, lines 46-50; column 43-54), when executed, operate the apparatus to:

initialize a file to store the web based application, including creation of a root directory within the file (See abstract; Fig. 2; column 1, lines 54-57; column 3, line 39);

initializing a first plurality of storage data objects under the data directories for all non-file system structures of the web based application (See abstract; column 2, lines 48-62, where “non-file system structures” is read on “tables”; column 3, lines 44-46); and

copy and store the non-file system structures into the first plurality of storage data objects (See abstract; Fig. 2 and Fig. 3; column 2, lines 48-62, where “non-file system structures” is read on “table”; column 3, lines 44-46; column 4, lines 53-62).

Byrne et al. does not teach create data directories under the root directory; and a processor coupled to the storage medium to execute the programming instructions.

Shadmon teaches a method and system for uniformly accessing multiple directory services (See abstract), in which he teaches create data directories under the root directory (See Fig. 1; column 38; lines 9-20, where “data dictionary” is read on “data directories”); and a processor coupled to the storage medium to execute the



programming instructions (See Fig. 1, where "Database Management system" is read on a "storage medium"; also see column 18, lines 50-52).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al., to include create data directories under the root directory; and a processor coupled to the storage medium to execute the programming instructions.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al., by the teachings of Shadmon because create data directories under the root directory; and a processor coupled to the storage medium to execute the programming instructions would maintain a so-called balanced structure of the tree index in order to avoid long paths for accessing a given data record from the root node to the leaf node that is associated with the sought data record (See Shadmon column 4, lines 38-42).

4. Claims 6-8, 16-18, 21, 23, 32 and 33-38 rejected under 35 U.S.C. 103(a) as being unpatentable over Byrne et al. (U.S. Patent No. 6,823,338) in view of Shadmon (U.S. Patent No. 6,208,993), as applied to claims 1, 3-5, 9, 11, 13-15 and 19 above, and further in view of Smith et al. (U.S. Patent No. 6,052,693).

As to claim 6 Byrne et al. as modified, teaches wherein the creating of data directories under the root directory and initializing a first plurality of storage data objects under the data directories (See Shadmon, Fig. 1; column 38; lines 9-20).

Art Unit: 2164

Byrne et al. as modified still does not teach, creating a plurality of data table directories under the application level data directory.

Smith et al. teaches a system for assembling large databases through information extracted from text sources (See abstract), in which he teaches creating a plurality of data table directories under the application level data directory (See column 33, lines 11-17).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. as modified, to include creating a plurality of data table directories under the application level data directory.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. as modified, by the teachings of Smith et al. because creating a plurality of data table directories under the application level data directory would describe each and every document, entity, and form and their interrelationships held in the application data tables which store the structured entity relationship model (See Smith et al., column 33, lines 24-27).

As to claim 7, Byrne et al. as modified, teaches wherein the creating of data directories under the root directory and initializing a first plurality of storage data objects under the data directories further comprises:

initializing a first subset of the first plurality of storage data objects under the data table directory to store data table schemas of the web based application (See

Art Unit: 2164

Smith et al., column 33, lines 11-17); and initializing a second subset of the first plurality of storage data objects under the data table directory to data tables of the web based application (See Smith et al., column 33, lines 11-17).

As to claims 8 and 18, Byrne et al. as modified, teaches wherein the copying and storing of non-file system structures into the first plurality of storage data objects comprises copying and storing data table schemas and data tables of the web based application into corresponding pairs of the first and second subset of the first plurality of storage data objects (See Smith et al., column 33, lines 11-17);

wherein the programming instructions, when executed, operate the apparatus to copy and store data table schemas and data tables of the web based application into corresponding pairs of the first and second subset of the first plurality of storage data objects to copy and store non-file system structures into the first plurality of storage data objects (See Smith et al., column 33, lines 11-17).

As to claim 16, Byrne et al. as modified, teaches wherein the programming instructions, when executed, operate the apparatus to create a plurality of data table directories under the application level data directory to create data directories under the root directory and initialize a first plurality of storage data objects under the data directories (See Smith et al., column 33, lines 11-17).

As to claim 17, Byrne et al. as modified, teaches wherein the programming instructions, when executed, operate the apparatus to:

initialize a first subset of the first plurality of storage data objects under the data table directory to store data table schemas of the web based application (See abstract; column 1, lines 54-57; column 3, line 39), and initialize a second subset of the first plurality of storage data objects under the data table directory to data tables of the web based application (See Smith et al., column 33, lines 11-17).

As to claim 21, Byrne et al. teaches a method for copying/restoring a web-based application into a domain (See abstract; column 1, lines 15-38);

retrieving a structural description describing non-file system structures and files of the web based application (See abstract; column 2, lines 48-62, where "non-file system structures" is read on "tables"; column 3, lines 44-46);

including constitutions of the non-file system structures, and files of the web-based application, including pathnames of the files (See abstract; column 1, lines 54-57; column 4, lines 15-30).

Byrne et al. does not teach, determining in accordance with at least the structural description non-file system structures of the web based application; retrieving schemas and data of the non-file system structures in accordance with the result of the determination storing the data of the non-file system structures in accordance with schemas of the non-file system structures; and retrieving and storing the files in accordance with the result of the determination.

Shadmon teaches a method and system for uniformly accessing multiple directory services (See abstract), in which he determining in accordance with at least the structural description non-file system structures of the web based application (See Fig. 1; column 38; lines 9-20, where "data dictionary" is read on "data directories").

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al., to include determining in accordance with at least the structural description non-file system structures of the web based application.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al., by the teachings of Shadmon because determining in accordance with at least the structural description non-file system structures of the web based application would make it easier and faster to find and retrieve relevant information to a user.

Byrne et al. as modified still does not teach, retrieving schemas and data of the non-file system structures in accordance with the result of the determination storing the data of the non-file system structures in accordance with schemas of the non-file system structures; and retrieving and storing the files in accordance with the result of the determination.

Smith et al. teaches a system for assembling large databases through information extracted from text sources (See abstract), in which he teaches retrieving schemas and data of the non-file system structures in accordance with the result of the determination storing the data of the non-file system structures in accordance with

Art Unit: 2164

schemas of the non-file system structures (See column 1, lines 16-25; column 4, lines 18-20; column 33, lines 11-17); and retrieving and storing the files in accordance with the result of the determination (See abstract; column 1, lines 41-49; column 6, lines 16-22; column 33, lines 11-17).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. as modified, to retrieving schemas and data of the non-file system structures in accordance with the result of the determination storing the data of the non-file system structures in accordance with schemas of the non-file system structures; and retrieving and storing the files in accordance with the result of the determination.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. as modified, by the teachings of Smith et al. because retrieving schemas and data of the non-file system structures in accordance with the result of the determination storing the data of the non-file system structures in accordance with schemas of the non-file system structures; and retrieving and storing the files in accordance with the result of the determination would describe each and every document, entity, and form and their interrelationships held in the application data tables which store the structured entity relationship model (See Smith et al., column 33, lines 24-27).

As to claim 23, Byrne et al. teaches an apparatus comprising:

Art Unit: 2164

a storage medium having stored therein a plurality of programming instructions (See abstract; column 43-54).

Byrne et al. does not teach at least one processor coupled to the storage medium to execute the programming instructions.

Shadmon teaches a method and system for uniformly accessing multiple directory services (See abstract), in which he teaches at least one processor coupled to the storage medium to execute the programming instructions (See Fig. 1, where "Database Management system" is read on a "storage medium"; also see column 18, lines 50-52).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. to include at least one processor coupled to the storage medium to execute the programming instructions.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. by the teachings of Shadmon because at least one processor coupled to the storage medium to execute the programming instructions would make it easier and faster to find and retrieve relevant information to a user.

Byrne et al. as modified, still does not teach when executed, operate the apparatus to retrieve a structural description describing non-file system structures and files of a web based application, determine in accordance with at least the structural description non-file system structures of the web-based application, including constitutions of the non-file system structures, and files of the web based application,

Art Unit: 2164

including pathnames of the files, retrieve schemas and data of the non-file system structures in accordance with the result of the determination, store the data of the non-file system structures in accordance with schemas of the non-file system structures, and retrieve and store the files in accordance with the result of the determination; and at least one processor coupled to the storage medium to execute the programming instructions.

Smith et al. teaches a system for assembling large databases through information extracted from text sources (See abstract), in which he teaches when executed, operate the apparatus to retrieve a structural description describing non-file system structures and files of a web based application, determine in accordance with at least the structural description non-file system structures of the web-based application (See column 1, lines 16-25; column 4, lines 18-20; column 33, lines 11-17), including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files, retrieve schemas and data of the non-file system structures in accordance with the result of the determination (See abstract; column 1, lines 41-49; column 6, lines 16-22; column 33, lines 11-17), store the data of the non-file system structures in accordance with schemas of the non-file system structures, and retrieve and store the files in accordance with the result of the determination (See abstract; column 1, lines 41-49; column 6, lines 16-22; column 33, lines 11-17).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. as modified, to include when executed, operate the apparatus to retrieve a structural description



Art Unit: 2164

describing non-file system structures and files of a web based application, determine in accordance with at least the structural description non-file system structures of the web-based application, including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files, retrieve schemas and data of the non-file system structures in accordance with the result of the determination, store the data of the non-file system structures in accordance with schemas of the non-file system structures, and retrieve and store the files in accordance with the result of the determination.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. as modified, by the teachings of Smith et al. because when executed, operate the apparatus to retrieve a structural description describing non-file system structures and files of a web based application, determine in accordance with at least the structural description non-file system structures of the web-based application, including constitutions of the non-file system structures, and files of the web based application, including pathnames of the files,

retrieve schemas and data of the non-file system structures in accordance with the result of the determination, store the data of the non-file system structures in accordance with schemas of the non-file system structures, and retrieve and store the files in accordance with the result of the determination would describe each and every document, entity, and form and their interrelationships held in the application data tables which store the structured entity relationship model (See Smith et al., column 33, lines 24-27).

As to claim 32 Byrne et al. teaches an apparatus comprising:  
a storage medium having stored therein a plurality of programming instructions  
(See abstract; column 14, lines 46-50; column 43-54).

Byrne et al. does not teach at least one processor coupled to the storage medium to execute the programming instructions.

Shadmon teaches a method and system for uniformly accessing multiple directory services (See abstract), in which he teaches at least one processor coupled to the storage medium to execute the programming instructions (See Fig. 1, where "Database Management system" is read on a "storage medium"; also see column 18, lines 50-52).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. to include at least one processor coupled to the storage medium to execute the programming instructions. It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. by the teachings of Shadmon because at least one processor coupled to the storage medium to execute the programming instructions would make it easier and faster to find and retrieve relevant information to a user.

Byrne et al. as modified, still does not teach the apparatus to retrieve a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables, as each data table schema is retrieved, store the data table schema

Art Unit: 2164

in a temporal storage location, create a data table in accordance with the data table schema, determine if data for the data table has already been retrieved, store the data into the data table if the data for the data table has already been retrieved, and as each collection of data for a data table is retrieved, store the collection of data, in a temporal storage location, determine if the data table has already been created, store the data into the data table if the data table has already been created.

Smith et al. teaches a system for assembling large databases through information extracted from text sources (See abstract), in which he teaches the apparatus to retrieve a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables, as each data table schema is retrieved (See column 33, lines 11-17), store the data table schema in a temporal storage location, create a data table in accordance with the data table schema (See column 33, lines 11-17), determine if data for the data table has already been retrieved (See column 33, lines 11-17; column 30, lines 43-51), store the data into the data table if the data for the data table has already been retrieved, and as each collection of data for a data table is retrieved, store the collection of data, in a temporal storage location, determine if the data table has already been created (See column 13, lines 29-30), and store the data into the data table if the data table has already been created.

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. as modified, to include the apparatus to retrieve a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables, as each data table

Art Unit: 2164

schema is retrieved, store the data table schema in a temporal storage location, create a data table in accordance with the data table schema, determine if data for the data table has already been retrieved, store the data into the data table if the data for the data table has already been retrieved, and as each collection of data for a data table is retrieved, store the collection of data, in a temporal storage location, determine if the data table has already been created, store the data into the data table if the data table has already been created.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. as modified, by the teachings of Smith et al. because the apparatus to retrieve a plurality of data table schemas for a plurality of data tables of a web based application, and data of the data tables, as each data table schema is retrieved, store the data table schema in a temporal storage location, create a data table in accordance with the data table schema, determine if data for the data table has already been retrieved, store the data into the data table if the data for the data table has already been retrieved, and as each collection of data for a data table is retrieved, store the collection of data, in a temporal storage location, determine if the data table has already been created, store the data into the data table if the data table has already been created would describe each and every document, entity, and form and their interrelationships held in the application data tables which store the structured entity relationship model (See Smith et al., column 33, lines 24-27).

Art Unit: 2164

As to claim 33, Byrne et al. as modified, teaches wherein the programming instructions, when executed, further operate the apparatus to delete the data table schema and the data of the data table stored in the respective temporal storage locations, upon storing the data of a data table into the data table (See Smith et al., column 33, lines 24-27).

As to claim 34, Byrne et al. as modified, teaches wherein the programming instructions, when executed, further operate the apparatus to delete log-in user names of users when storing data into a data table if the data table is an address book (See Smith et al., column 4, lines 3-8; column 33, lines 24-27).

As to claim 35, Byrne et al. as modified, teaches wherein the programming instructions, when executed, further operate the apparatus to determine if users having entries in an address book are authorized to log in the domain, and add into corresponding entries of the address book log-in user names of users authorized to log in the domain (See Byrne et al., abstract; column 5, lines 41-53; also see Smith et al., column 4, lines 3-8; column 33, lines 24-27).

As to claim 36, Byrne et al. as modified, teaches wherein the programming instructions, when executed, further operate the apparatus to conditionally delete or retain log-in user names of users depending on whether the users are authorized to log in the domain when storing data into a data table if the data table is an address book

Art Unit: 2164

(See Byrne et al., abstract; column 5, lines 41-53; also see Smith et al., column 33, lines 24-27).

As to claim 37, Byrne et al. as modified, teaches wherein the programming instructions, when executed, further operate the apparatus to:

retrieve a list of users of the web based applications (See Byrne et al., column 5, lines 59-61; column 16, lines 22-26; column 233, lines 14-17, lines 23-25); determine if the users are registered with the domain (See Byrne et al., abstract; column 5, lines 41-53); and register the users with the domain if the users are determined to be not having registered with the domain (See Byrne et al., abstract; column 5, lines 41-53).

As to claim 38, Byrne et al. as modified, teaches wherein the programming instructions, when executed, further operate the apparatus to:

determine if the users already have corresponding entries in an address book of the web based application (See Byrne et al., abstract; column 5, lines 41-53); create the corresponding entries in the address book if the corresponding entries are determined not to have been previously created (See Smith et al., column 30, lines 43-51); and upon either determining the existence or creation of the corresponding entries, updating the corresponding entries with log-in user names of the users (See Byrne et al., abstract; column 5, lines 41-53).

Art Unit: 2164

5. Claims 25-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Byrne et al. (U.S. Patent No. 6,823,338) in view of Smith et al. (U.S. Patent No. 6,052,693).

As to claim 25, Byrne et al. as modified, teaches a method for copying/restoring a web based application into a domain (See Byrne et al., abstract; column 1, lines 15-38).

Byrne et al. as modified still does not teach retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables; as each data table schema is retrieved, storing the data table schema in a temporal storage location, creating a data table in accordance with the data table schema, determining if data for the data table has already been retrieved, storing the data into the data table if the data for the data table has already been retrieved; and as each collection of data for a data table is retrieved, storing the collection of data in a temporal storage location, determining if the data table has already been created, storing the data into the data table if the data table has already been created.

Smith et al. teaches a system for assembling large databases through information extracted from text sources (See abstract), in which he teaches retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables (See column 33, lines 11-17);

as each data table schema is retrieved, storing the data table schema in a temporal storage location (See column 13, lines 29-30; column 33, lines 11-17), creating a data table in accordance with the data table schema, determining if data for the data table has already been retrieved (See column 30, lines 43-51), storing the data

Art Unit: 2164

into the data table if the data for the data table has already been retrieved (See column 33, lines 11-17); and

as each collection of data for a data table is retrieved, storing the collection of data in a temporal storage location, determining if the data table has already been created, storing the data into the data table if the data table has already been created (See column 13, lines 29-30; column 33, lines 11-17).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al., to include retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables; as each data table schema is retrieved, storing the data table schema in a temporal storage location, creating a data table in accordance with the data table schema, determining if data for the data table has already been retrieved, storing the data into the data table if the data for the data table has already been retrieved; and as each collection of data for a data table is retrieved, storing the collection of data in a temporal storage location, determining if the data table has already been created, storing the data into the data table if the data table has already been created.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al., by the teachings of Smith et al. because retrieving a plurality of data table schemas for a plurality of data tables of the web based application, and data of the data tables; as each data table schema is retrieved, storing the data table schema in a temporal storage location, creating a data



Art Unit: 2164

table in accordance with the data table schema, determining if data for the data table has already been retrieved, storing the data into the data table if the data for the data table has already been retrieved; and as each collection of data for a data table is retrieved, storing the collection of data in a temporal storage location, determining if the data table has already been created, storing the data into the data table if the data table has already been created would describe each and every document, entity, and form and their interrelationships held in the application data tables which store the structured entity relationship model (See Smith et al., column 33, lines 24-27).

As to claim 26, Byrne et al. as modified, teaches wherein the method further comprises upon storing the data of a data table into the data table (See Smith et al., column 33, lines 24-27), deleting the data table schema and the data of the data table stored in the respective temporal storage locations (See Smith et al., column 33, lines 24-27).

As to claim 27, Byrne et al. as modified, teaches wherein the method further comprises deleting log-in user names of users when storing data into a data table if the data table is an address book (See Smith et al., column 4, lines 3-8; column 33, lines 24-27).

As to claim 28, Byrne et al. as modified, teaches wherein the method further comprises determining if users having entries in an address book are authorized to log

Art Unit: 2164

in the domain, and adding into corresponding entries of the address book log-in user names of users authorized to log in the domain (See Byrne et al., Byrne et al., abstract; column 1, lines 22-26, lines 28-59; column 5, lines 40-53; column 6, lines 24-28; column 8, lines 1-8; column 10, lines 51-52; also see Smith et al., column 4, lines 3-8; column 33, lines 24-27).

As to claim 29, Byrne et al. as modified, teaches wherein the method further comprises conditionally deleting or retaining log-in user names of users depending on whether the users are authorized to log in the domain when storing data into a data table if the data table is an address book (See Byrne et al., abstract; column 1, lines 22-26, lines 28-59; column 5, lines 40-53; column 6, lines 24-28; column 8, lines 1-8; column 10, lines 51-52; also see Smith et al., column 33, lines 24-27).

As to claim 30, Byrne et al. as modified, teaches wherein the method further comprises:

retrieving a list of users of the web based applications (See Byrne et al., abstract; column 1, lines 28-59; column 5, lines 40-53; column 10, lines 51-52); determining if the users are registered with the domain (See Byrne et al., abstract; column 1, lines 22-26, lines 28-59; column 5, lines 40-53; column 6, lines 24-28; column 8, lines 1-8; column 10, lines 51-52); and registering the users with the domain if the users are determined to be not having registered with the domain (See Byrne et al., abstract; column 1, lines 28-59; column 5, lines 40-53; column 10, lines 51-52).

As to claim 31, Byrne et al. as modified, teaches wherein the method further comprises:

determining if the users already have corresponding entries in an address book of the web based application (See Byrne et al., abstract; column 1, lines 28-59; column 5, lines 40-53; column 10, lines 51-52); creating the corresponding entries in the address book if the corresponding entries are determined not to have been previously created (See Smith et al., column 30, lines 43-51); and upon either determining the existence or creation of the corresponding entries, updating the corresponding entries with log-in user names of the users (See Byrne et al., abstract; column 1, lines 28-59; column 5, lines 40-53; column 10, lines 51-52).

6. Claims 2 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Byrne et al. (U.S. Patent No. 6,823,338) in view of Shadmon (U.S. Patent No. 6,208,993), further in view of Smith et al. (U.S. Patent No. 6,052,693), as applied to claims 1, 3-5, 9, 11, 13-15 and 19 above, and further in view of Bodin et al. (U.S. Patent No. 6,604,106).

As to claim 2, Byrne et al. as modified still does not teach wherein the initializing of a file to store the web-based application comprises initializing a compressible file.

Bodin et al. teaches compression and delivery of web server content (See abstract), in which he teaches wherein the initializing of a file to store the web based application comprises initializing a compressible file (See column 4, lines 18-25).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. as modified, to include wherein the initializing of a file to store the web based application comprises initializing a compressible file.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. as modified, by the teachings of Bodin et al. because wherein the initializing of a file to store the web based application comprises initializing a compressible file would minimize the amount of storage space need to store the file.

As to claim 12, Byrne et al. as modified, teaches wherein the programming instructions, when executed, operate the apparatus to initialize a compressible file to store the web-based application (See Bodin et al., column 4, lines 18-25).

7. Claims 10, 20, 22 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Byrne et al. (U.S. Patent No. 6,823,338) in view of Shadmon (U.S. Patent No. 6,208,993) and further in view of Smith et al. (U.S. Patent No. 6,052,693), as applied to claims 1, 3-5, 9, 11, 13-15 and 19 above, and further in view of Gai et al. (U.S. Patent No. 6,651,096).

As to claim 10, Byrne et al. as modified, teaches wherein the copying and storing of files of the web based application (See Byrne et al., objects (See abstract; Fig. 2 and Fig. 3; column 2, lines 48-62, where "non-file system structures" is read on "table";

Art Unit: 2164

column 3, lines 44-46; column 4, lines 53-62), that are part of a file system into the file for storing the web based application as second plurality of storage data objects under the root directory (See Byrne et al., abstract; Fig. 2; column 1, lines 54-57; column 3, line 39).

Byrne et al. as modified, still does not teach pre-processing access control lists into a self-describing format before storing the access control lists into selected ones of the second plurality of storage data objects.

Gai et al. teaches a method and apparatus for organizing, storing and evaluation access control lists (See abstract), in which pre-processing access control lists into a self-describing format before storing the access control lists into selected ones of the second plurality of storage data objects (See abstract; column 14, lines 56-62; column 20, lines 30-34).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time of the invention was made to have modified Byrne et al. as modified, to include pre-processing access control lists into a self-describing format before storing the access control lists into selected ones of the second plurality of storage data objects.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Byrne et al. as modified, by the teachings of Gai et al. pre-processing access control lists into a self-describing format before storing the access control lists into selected ones of the second plurality of storage data objects would optimize the creation and evaluation of multiple access control lists so as to

Art Unit: 2164

maintain, if not improve, packet processing speeds (See Gai et al., column 3, lines 34-36).

As to claim 20, Byrne et al. as modified, teaches wherein the programming instructions, when executed, operate the apparatus to pre-process access control lists into a self describing format before storing the access control lists into selected ones of the second plurality of storage data objects (See Byrne et al., abstract; column 4, lines 66-67; column 5, lines 1-16; also see Gai et al., abstract; column 14, lines 56-62; column 20, lines 30-34).

As to claim 22, Byrne et al. as modified, teaches wherein the retrieving and storing of files of the web based application (See Byrne et al., (See abstract; Fig. 2 and Fig. 3; column 2, lines 48-62; column 3, lines 44-46; column 4, lines 53-62) comprises transforming one or more access control lists into a binary format before storing the one or more access control lists (See Gai et al., abstract; column 9, lines 25-27; column 20, lines 30-34).

As to claim 24, Byrne et al. as modified, teaches wherein the programming instructions, when executed, further operate the apparatus; to transform a access control list into a binary format before storing the access control list (See Gai et al., abstract; column 9, lines 25-27; column 20, lines 30-34).

***Response to Arguments***

8. Applicant's arguments filed on 18-April-2005, with respect to the rejected claims 1-38 have been fully considered but they are not found to be persuasive:

In response to applicants' arguments regarding Byrne does not show "a file to store said web based application...including...a root directory with in said file" the argument has been fully considered but are not found to be persuasive, because Byrne discloses "ACL" data stored in a relational database using a plurality of "tables". Byrne further discloses "LDAP directory" which is provided through ACL's (Access control lists), (See column 1, lines 58-67; column 2, lines 1-2, lines 48-62).

In response to applicants' arguments regarding Byrne does not teach or suggest "a file directory structure 'within' the file to copy/archive the web-base application with both 'file system' and 'non-file system' structures" the argument has been fully considered but are not found to be persuasive, because Byrne discloses "ACL" data stored in a relational database using a plurality of "tables". Byrne further discloses "LDAP directory" which is provided through ACL's (Access control lists), (See column 1, lines 58-67; column 2, lines 1-2, lines 48-62). Shadmon discloses a "file management system for accessing data records that correspond to items in a directory" (See abstract). Combining Byrne's "ACL data stored in a relational database" with the file management system of Shadmon would disclose "a file directory structure 'within' the file to copy/archive the web-base application with both 'file system' and 'non-file system' structures".

***Conclusion***

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mellissa M. Chojnacki whose telephone number is (571) 272-4076. The examiner can normally be reached on 9:00am-5:30pm.


If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.



Art Unit: 2164

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

July 8, 2005  
Mmc

  
**CHARLES RONES**  
**PRIMARY EXAMINER**